**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Рязанский государственный радиотехнический университет имени В.Ф. Уткина»

Кафедра «Электронные вычислительные машины»

**Отчет о лабораторной работе № 10**

по теме

«Java DataBase Connectivity»

**Выполнили:**

Студенты гр. 045

Бригады №10

Анохин В.А.

Вашкулатов Н.А.

**Проверил:**

Асс. Бастрычкин А.С.

Цель работы:изучение взаимодействия языка программирования Java с базами данных, приобретение навыков работы с БД в Java-программах.

Ход работы

Задание

Города. В БД хранится информация о городах и их жителях.

Для городов необходимо хранить:

• название;

• год основания;

• площадь;

• количество населения для каждого типа жителей.

Для типов жителей необходимо хранить:

• город проживания;

• название;

• язык общения.

• Вывести информацию обо всех жителях заданного города, разговаривающих на заданном языке.

• Вывести информацию обо всех городах, в которых проживают жители выбранного типа.

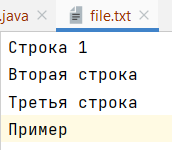
• Вывести информацию о городе с заданным количеством населения и всех типах жителей, в нем проживающих.

• Вывести информацию о самом древнем типе жителей.

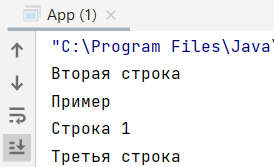
Код программы:

import java.io.FileReader;  
import java.io.IOException;  
import java.util.ArrayList;  
import java.util.Scanner;  
  
public class App {  
 public static void main(String[] args) {  
 final String path = "C:\\Users\\Вячеслав\\Desktop\\labs3Turn\\Java\\lab\_9\\src\\Ex1\\file.txt";  
 ArrayList<String> lines = new ArrayList<>();  
 try(Scanner scan = new Scanner(new FileReader(path))) {  
 while (scan.hasNextLine()) {  
 lines.add(scan.nextLine());  
 }  
 }  
 catch (IOException e){  
 System.err.println(e);  
 }  
 lines.sort(String::compareTo);  
 lines.forEach(System.out::println);  
 }  
}

Текстовый файл:



Окно вывода:



Задание 2

Один из способов шифрования данных, называемый «двойным шифрованием», заключается в том, что исходные данные при помощи некоторого преобразования последовательно шифруются на некоторые два ключа — K1 и K2. Разработать и реализовать эффективный алгоритм, позволяющий находить ключи K1 и K2 по исходной строке и ее зашифрованному варианту. Проверить, оказался ли разработанный способ действительно эффективным, протестировав программу для случая, когда оба ключа являются 20-битными (время ее работы не должно превосходить одной минуты).

Код программы:

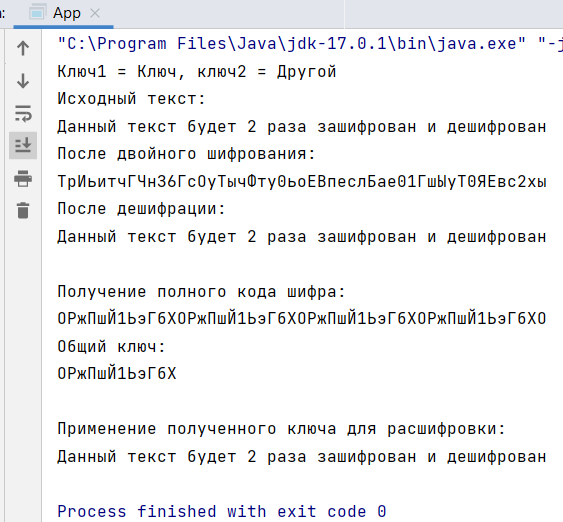
Класс VigenerCipher

import java.util.ArrayList;  
  
public class VigenerCipher {  
 ArrayList<Character> alphabet;  
  
 VigenerCipher(){  
 alphabet = new ArrayList<>(75);  
 for(char c = 'А'; c<='я'; ++c){  
 alphabet.add(c);  
 }  
 alphabet.add(' ');  
 for(char c='0'; c<='9'; ++c){  
 alphabet.add(c);  
 }  
 }  
  
 public String encode(String text, String key) {  
 StringBuilder res = new StringBuilder(text.length());  
 for(int i=0; i<text.length(); ++i){  
 res.append(getCipherChar(text.charAt(i),key.charAt(i % key.length())));  
 }  
 return res.toString();  
 }  
  
 public String decode(String text, String key) {  
 StringBuilder res = new StringBuilder(text.length());  
 for(int i=0; i<text.length(); ++i){  
 res.append(getSourceChar(text.charAt(i),key.charAt(i% key.length())));  
 }  
 return res.toString();  
 }  
  
 private char getCipherChar(char source, char shift){  
 return alphabet.get((alphabet.indexOf(source) + alphabet.indexOf(shift)) % alphabet.size());  
 }  
  
 private char getSourceChar(char cipher, char shift){  
 return alphabet.get((alphabet.indexOf(cipher) - alphabet.indexOf(shift) + alphabet.size()) % alphabet.size());  
 }  
  
 public String getNonParsedKey(String text, String encoded){  
 StringBuilder res = new StringBuilder(text.length());  
 for(int i=0; i<text.length(); ++i){  
 res.append(getSourceChar(encoded.charAt(i), text.charAt(i)));  
 }  
 return res.toString();  
 }  
  
 public String getLongKey(String nonParsedKey){  
 int keyEndPos, nextKeyPos = 1;  
 String key;  
 while (true){  
 key = nonParsedKey.substring(0, nextKeyPos);  
 keyEndPos = nextKeyPos;  
 nextKeyPos = nonParsedKey.indexOf(key, keyEndPos);  
 if(keyEndPos == nextKeyPos){ //Проверка правильности ключа  
 int keyLen = key.length();  
 for(int i=0; i<nonParsedKey.length()-keyLen+1; i+=keyLen){  
 String tmp = nonParsedKey.substring(i, i+keyLen);  
 if(!tmp.equals(key)){  
 nextKeyPos++;  
 break;  
 }  
 }  
 if(keyEndPos == nextKeyPos){  
 break;  
 }  
 }  
 if(nextKeyPos == -1){ //Выполняется, если дано полтора ключа  
 String endKey = nonParsedKey.substring(nonParsedKey.length()-1);  
 int endLen = nonParsedKey.indexOf(endKey)+1;  
 endKey = nonParsedKey.substring(0, endLen);  
 int last;  
 do {  
 last = nonParsedKey.lastIndexOf(endKey);  
 endKey = nonParsedKey.substring(last-1);  
 endLen = nonParsedKey.indexOf(endKey)+1;  
 } while (last != endLen);  
 key = nonParsedKey.substring(0, last);  
 break;  
 }  
 }  
 return key;  
 }  
}

Основная программа:

public class App {  
 public static void main(String[] args) {  
 String text = "Данный текст будет 2 раза зашифрован и дешифрован";  
 String key = "Ключ";  
 String key2 = "Другой";  
 VigenerCipher cipher = new VigenerCipher();  
 String encoded = cipher.encode(text, key);  
 String encoded2 = cipher.encode(encoded, key2);  
 String decoded = cipher.decode(encoded2, key);  
 String decoded2 = cipher.decode(decoded, key2);  
 System.out.println("Ключ1 = "+key+", ключ2 = "+key2);  
 System.out.println("Исходный текст:");  
 System.out.println(text);  
 System.out.println("После двойного шифрования:");  
 System.out.println(encoded2);  
 System.out.println("После дешифрации:");  
 System.out.println(decoded2);  
 System.out.println();  
 System.out.println("Получение полного кода шифра:");  
 String nonParsedKey = cipher.getNonParsedKey(text, encoded2);  
 System.out.println(nonParsedKey);  
 System.out.println("Общий ключ:");  
 String longKey = cipher.getLongKey(nonParsedKey);  
 System.out.println(longKey);  
 System.out.println();  
 System.out.println("Применение полученного ключа для расшифровки:");  
 System.out.println(cipher.decode(encoded2, longKey));  
 }  
}

Окно вывода:



**Вывод:** мы изучили коллекции Java, реализующие различные алгоритмы и структуры данных, приобрели навыки использования коллекций в Java-программах.